```
stack = []

exp = input("Enter postfix expression: ")

for ch in exp:
    if ch.isdigit():
        stack.append(int(ch))
    else:
        b = stack.pop()
        a = stack.pop()

        if ch == '+':
            stack.append(a + b)
        elif ch == '-':
            stack.append(a - b)
        elif ch == '*':
            stack.append(a * b)
        elif ch == '/':
            stack.append(a // b)

print("Result:", stack.pop())
```

Output:

```
Enter postfix expression: 23+
Result: 5
```

INFIX:

```
def precedence(op):
    if op in ('+', '-'):
        return 1
    if op in ('*', '/'):
        return 2
    return 0

def apply_operator(op, b, a):
    if op == '+':
        return a + b
    if op == '-':
        return a - b
    if op == '*':
        return a * b
    if op == '/':
        return a / b

def evaluate_infix(expression):
    operands = []
    operators = []

    for ch in expression:
        if ch.isdigit():
            operands.append(int(ch))

        elif ch in "+-*/":
            while (operators and
                   precedence(operators[-1]) >= precedence(ch)):
                op = operators.pop()
                b = operands.pop()
                a = operands.pop()
                operands.append(apply_operator(op, b, a))
            operators.append(ch)

    while operators:
        op = operators.pop()
        b = operands.pop()
        a = operands.pop()
        operands.append(apply_operator(op, b, a))

    return operands[0]

exp = input("Enter infix expression: ")
print("Total =", evaluate_infix(exp))
```

OUTPUT:

```
Enter infix expression: 2+3*4
Total = 14
```